

while the other three states reflect the additional functionality of a STAR bridge. When the spanning tree is being built, each bridge (both old and STAR bridges) is in the Tree Learning state. In this state, all data frames are dropped as in the IEEE 802.1D standard. After the spanning tree is built, the bridge goes to the Tree Learned state. A path finding process starts and data frames are forwarded using the standard forwarding and learning processes. There are several phases in the path finding process. The bridge first goes to the Distant Neighbors Found state, then the Direct Neighbors Found state, and finally the Enhanced state in different phases. The transitions will be explained further hereinafter in the Path Finding Process section. When the bridge is in the Enhanced state, the path finding process is completed. STAR forwarding and learning processes are executed when a data frame is received in this state.

I.B. Port States

In the IEEE 802.1D standard, there are four port states: Blocking, Listening, Learning, and Forwarding. In the STAR Bridge Protocol, there are three additional port states for the path finding process, which are similar to the Listening, Learning, and Forwarding states. These port states, and the transactions among the port states are shown in FIG. 3. These new port states are distance vector listening (DV Listening), distance vector learning (DV Learning), and STAR Forwarding. The IEEE 802.1D Spanning Tree Bridge Protocol activates tree ports while the path finding process activates other useful non-tree ports. The transition among the four states in the standard remains the same for the STAR bridges. A port changes from the Blocking state to the Listening state when it is selected as a tree port. After an appropriate protocol timer expires, the port moves to the Learning state. In this state, learning is enabled but data frames will not be forwarded. The port is in the Forwarding state if the timer expires again. A port changes from the Blocking state to the DV Listening state if the path finding process selects it. A protocol timer is started when the port enters the DV Listening state. When this timer expires, the port moves to the DV Learning state. Unlike the Learning state in the present IEEE standard, in which a port learns the locations of end stations, a port in DV Learning state does not do this. It is because all data frames are still forwarded on tree paths and a port in DV Learning state must be a non-tree port. Hence, there should be no data frame arriving on that port. A similar protocol timer is used to time out the DV Learning state, and the next transition is into the Forwarding state.

I.C. Storage

Each bridge keeps an FD for the forwarding process. In STAR bridges, three tables are additionally used provided: *bridge forwarding table (BF Table)*, *end-station location table (ESL Table)*, and *bridge address table (BA Table)*. These tables are preferably stored in memory devices already resident in non-STAR bridges, such as memory for storing FD. A BF Table indicates for each other STAR bridge in the bridged LAN the port that leads to the next hop along the "best" path found. BF Tables are obtained in the path finding process by a modified *distance vector* method as will be described in further detail hereinafter. An ESL Table is used to map an end station to a STAR bridge near it. The STAR learning process is responsible for filling this table. Therefore, if the ESL Table of a STAR bridge has a record for an end station, it is unnecessary for the FD of the STAR bridge to have a record for the same

end station in most situations. This implies that the FD in a STAR bridge is no larger than that in a standard bridge. The BA Table is a mapping between STAR bridge identifiers and STAR bridge MAC addresses. Every bridge has its own unique MAC address and this MAC address is used in the STAR forwarding process. It is not necessary for a STAR bridge to know the MAC address of all other STAR bridges. It should be noted that $addr(n)$ will be the MAC address of a bridge with a bridge ID n .

I.D. Protocol Data Units

The STAR Bridge Protocol recognizes two types of protocol data units, namely BPDU (Bridge Protocol Data Units), which is specified in the IEEE 802.1D standard, and SBPDU (STAR Bridge Protocol Data Units), which are specified below for the STAR Bridge Protocol.

The SBPDU contains an SBPDU Header, which has the same format as the BPDU Header, and a set of SBPDU Parameters. The Protocol Identifier in the SBPDU has its own unique value to identify the STAR Bridge Protocol. SBPDU MAC frames assume a format similar to that of BPDU MAC frames. There are four kinds of SBPDU frames: Hello SBPDU, Distance Vector Change Notification SBPDU (DVCN_SBPDU), Distance Vector Computation SBPDU (DVC_SBPDU), and Station Location Announcement SBPDU (SLA_SBPDU). Hello SBPDU is used for monitoring crosslink failures. DVCN_SBPDU is used to notify STAR bridges about topology changes. DVC_SBPDU is used in the path finding process to compute the distance vectors and BF Tables. SLA_SBPDU is generated in the STAR learning process to fill the ESL Tables.

Since any BPDU frame with an unknown Protocol Identifier will not be forwarded by old bridges, an SBPDU frame sent by a STAR bridge must be encapsulated as a data frame if it is expected to traverse at least one old bridge. The source address of an encapsulated SBPDU frame is the MAC address of the STAR bridge that is responsible for the encapsulation. The destination MAC address is either the STAR bridge group MAC address or the unique MAC address of the intended STAR bridge recipient. Which destination MAC address is applicable depends on the specific type of SBPDU MAC frame being sent. In the proposed protocol, most SBPDU MAC frames are sent over the spanning tree. It will be explained herein that some SBPDU MAC frames may be sent to direct STAR neighbors over selected non-tree links. Except in the only case described in with reference to crosslink maintenance and Path Finding Process, an SBPDU MAC frame received by a STAR bridge will not be forwarded by the STAR bridge.

I.E. Data Frames

Each data frame at the LLC Sublayer contains LLC information and is encapsulated within a MAC frame at the MAC layer of the end station. We refer a MAC data frame generated by an end station as a normal MAC data frame, or simply a normal data frame when the context is clear. As will be explained in greater detail hereinafter with reference to the frame dropping problem, a normal data frame may have to be encapsulated by a bridge for forwarding purpose. When a normal MAC data frame is encapsulated, we refer to it as an encapsulated MAC data frame, or simply an encapsulated data frame when the context is clear. We refer to the intended recipient of an encapsulated data frame as a proxy destination, and its address as a proxy destination address. In the rest of this document, the phrase "data frame" refers to either a normal data frame or an encapsulated data frame. We assume that a MAC data frame *fr* has the form $\langle src(fr), dst(fr), pid(fr) \rangle$ where *src(fr)*, *dst(fr)*, and *pid(fr)* represent the source MAC address, the destination MAC address, and the payload data respectively.

A normal MAC data frame consists of a MAC Header, an LLC Header, a layer 3 packet, and a MAC Trailer. When a normal MAC data frame is encapsulated, an additional MAC Header and LLC header are put in the front of the normal data frame and an additional MAC Trailer is put at the end as shown in FIG. 4. Each STAR bridge has to distinguish among the following frames: normal data frames, encapsulated data frames, and encapsulated SBPDU frames. All these frames appear as normal data frames to old bridges. The present invention specifies frame type information in the Protocol Type field of the encapsulating LLC header so that a STAR bridge can correctly identify the frame type and process the frame.

In most bridged LANs, frames are subject to a maximum transfer unit (MTU) constraint. An encapsulated MAC frame of a given size may have to be fragmented before encapsulation, or the resulting frame will violate the MTU constraint. In this respect, STAR bridges must implement a fragmentation and reassembly mechanism to accommodate the encapsulation that may be needed for forwarding certain data frames. It follows that each encapsulated frame must include an additional field to carry an appropriate sequence number. It is sufficient to fragment an oversized frame into two fragments since the encapsulation overhead is much less than an MTU. Hence, only one bit is needed to identify a pair of fragments. Fragmentation and reassembly mechanisms are beyond the scope of the present invention since, frame encapsulation is already an existing function in present day bridges.

I.F. Crosslink Maintenance

Since a crosslink is not part of the active topology of a spanning tree, topology changes that involve crosslinks normally will not trigger a bridge to send out a Topology Change Notification BPDU. Therefore, a mechanism is needed for monitoring and updating the status of each crosslink selected by the proposed protocol for supporting an enhanced forwarding path. Hello SBPDUs and DVCN_SBPDU are used for this purpose. A Hello SBPDU consists merely of a Protocol Identifier field, a Protocol Version Identifier field, and an SBPDU Type field with a code reserved for this type. A DVCN_SBPDU contains an SBPDU Header and IDs of both bridges on the ends of the crosslink that is being identified to have failed.